1019-1002

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of: Fultheim, Shai	Application No.: 10/828,465
Filed: 04/21/04	Art Unit: 2128
For: Cluster Based Operating System-Agnostic Virtual Computing System	Examiner: David Silver

DECLARATION UNDER 37 CFR 1.132

I, the undersigned, Joseph Landman, hereby declare as follows:

1) I am making this Declaration in support of the patentability of the claims in U.S. Patent Application 10/828,465 (referred to hereinafter as "the Application"). Specifically, this Declaration will set forth my opinion, with supporting facts and evidence, that:

(a) The references that the Examiner has cited against the claims in the Application could not have led a person of ordinary skill in the art to make the claimed invention; and

(b) The invention defined by the claims in the Application, embodied in the vSMP product sold by ScaleMP (the assignee of the Application), answers a market need that could not be satisfied by prior art solutions and has been enthusiastically received by market leaders and customers.

2) I am not an employee of ScaleMP and have no economic interest in the company. I have not received compensation for my services in preparing this Declaration.

3) I have worked in the computer industry for more than 20 years, specializing in high-performance computing (HPC). I received my B.A. in physics in from the State University of New York at Stony Brook in 1987, my M.S. in physics from Michigan State University in 1990, and my Ph.D. in computational physics from Wayne State University in 1997. I have worked as an engineer., a researcher at a number of leading computer and software companies, including IBM and Silicon Graphics. I have also served as a part time research assistant professor in the computer science department at Wayne State University, teaching high performance computing programming. In 2002 I founded my own company, Scalable Informatics, which provides solutions for HPC users, including supercomputers, clusters, and storage systems. My detailed *curriculum vitae* is attached hereto as Exhibit A.

4) I am considered to be an industry authority in the field of HPC and have a detailed, first-hand understanding of both the technical and marketing aspects of this field. I publish a widely-read blog on HPC at scalability.org. I have given talks at various HPC and scientific venues over the last 12 years, including Linux World, ISMB, GSAC, and AINA. I have written white papers under contract to multiple organizations surrounding topics in HPC systems and applications and have published peer-reviewed papers in ACM and IEEE journals. I have also collaborated in development of a number of new HPC systems and programming tools, as detailed in my *curriculum vitae*.

5) ScaleMP has developed a software technology named vSMP, which is described and claimed in the Application. This

technology provides a shared virtual machine across multiple independent physical machines. vSMP aggregates resources of the underlying physical hardware, providing a single process space, memory address space, and I/O space for an operating system. As a result, multiple physical machines operate as a single virtual machine with the aggregated number of CPUs, memory size, and I/O space. This is a different, novel, and unique form of virtualization of resources, relative to methods that were previously known in the art.

6) Virtualization techniques prior to the development of vSMP subdivided a single computer into multiple smaller computers, providing instruction set emulation if required, as well as I/O virtualization. Such techniques replace direct access to the underlying hardware with a simplified virtual device that a "guest" operating system can use. This form of virtualization, as described in U.S. Patent 6,075,938 and in the "Disco" article cited against the Application (both by Bugnion), thus subdivides a single machine into many smaller "virtual machines," which have some number of virtual processors, some amount of memory, and some access to I/O and network resources. This is precisely the type of virtualization currently used in VMWare and similar products, and it is diametrically opposite in functionality and intention to vSMP.

7) The methods of virtualization described by Bugnion and implemented in VMWare do not allow the individual virtual machines to share memory, in the sense that for any two of these virtual machines, we cannot run a single process that consumes the sum of all memory across these machines. They also do not allow the

individual virtual machines to share processors, in the sense that the operating system on running on one virtual machine cannot directly schedule work on the second virtual machine. vSMP provides these capabilities, and does so transparently, permitting a standard operating system, without change, to run over multiple separate and independent computers.

8) The two different forms of virtualization I have described above - Bugnion's virtualization (VMWare) as opposed to that described in the Application (vSMP) - are orthogonal to each other. By "orthogonal," I mean that it is not an obvious or natural step to obtain the one form from the other form. Specifically, while it may be natural to use certain techniques and methods to subdivide one machine into many machines, it is unnatural to use the techniques of subdivision to aggregate multiple machines. While both forms use the term "virtual machine," the term is being used in a different technological context in each case. A "virtual machine" is not necessarily a physical machine to be sure, but it has a very different meaning depending upon which virtualization method is in use.

9) Moreover, the use of the term "virtual machine" in US Patent Application Publication 2003/0005068 (Nickel) is orthogonal to both of these concepts. Nickel relates to the venerable Parallel Virtual Machine (PVM) system developed decades ago, which has nothing to do with either of the virtualization concepts discussed above. The term "Virtual Machine" is, in the PVM context, a descriptive name, not a set of techniques to create multiple machine instances out of a single machine instance (as in VMWare), nor a set of techniques to create a seamless single

system image out of many machines, enabling all machines to transparently use all resources of each machine (as in vSMP). PVM is simply a programming framework for application-level programming, which allows single or multiple processes on each independent machine to communicate in a predefined manner. This framework allows the programmer to create a distributed algorithm for computing, using a number of networked machines, each running its own processes.

10) In the context of PVM programming, the term "virtual machine" refers to the process or processes that the program creates using the independent operating system on each machine, enabling the programmer to send data back and forth between the machines in a well defined manner. A PVM of the type described by Nickel is incapable of running an operating system and can run only programs that have been suitably modified. Despite using the term "virtual machine," Nickel's form of "virtualization" is entirely different from both VMWare and vSMP.

11) The following table summarizes the differences between the three types of "virtualization" that I described above:

Feature	**VMWare**	**vSMP**	**PVM**
Transparently use all processors on multiple component machines	No	Yes	No
Transparently use all memory on multiple component machines	No	Yes	No
Transparently use all networking resources on multiple component machines	No	Yes	No

Make one component machine appear as more than one machine	Yes	No	No
Enable unaltered programs to run utilizing resources	Yes (in most cases)	Yes	No
Enable unaltered operating systems to run utilizing resources	Yes (in most cases)	Yes	No
Enable large machine count single system images	No	Yes	No
Enable distributed programs to run in processes	Yes	Yes	Yes (must be written to PVM API)

It can be seen in the table above that aside from their use of the term "virtual machine," the three technologies have little in common. PVM seeks only to solve the bare-minimum issues of how to launch a collection of N distributed processes and how to provide a communication layer between them. Unlike vSMP, VMWare seeks to subdivide resources, not aggregate them, and to hide full system resources from operating systems, rather than exposing them.

12) For these reasons, a programmer in the field of multiprocessor computing - the "person having ordinary skill in the art" with respect to the Application - would never have even thought to apply the principles of process communication used in PVM in order to modify a VMWare virtual machine monitor. PVM and VMWare address different types of problems in ways that are mutually orthogonal, as I have explained above. Collaboration of multiple, loosely-coupled physical machines by PVM runs diametrically against the VMWare principle of subdividing a single physical machine into multiple virtual machines.

13) Furthermore, even if such a programmer had been "inspired" by PVM to try to modify the VMWare software in an attempt to run a single virtual machine over multiple physical machines, he still would have had no idea how to do so. As the table above shows, vSMP has a range of features that are outside the scope of capabilities of either VMWare or PVM. Solving the problems related to implementing these features required the inventors of vSMP to make a number of non-obvious inventive steps, as explained and claimed in the Application. The prior art does not provide the teachings that would have been needed by the person of ordinary skill in order to make these steps.

14) In the context of multi-computer systems, vSMP solves a specific set of problems that the other forms of virtualization do not address (and in some cases even exacerbate). By aggregating resources to provide a single system image over across multiple physical machines, vSMP provides a single point of management, a single operating system, and a single I/O space. This allows for:

- A reduction in system management costs and complexity, with only one "machine" to manage.
- A reduction in cost of operating system licenses: only one license required, rather than many.
- A simplified operating environment, enabling programs to operate in parallel without any additional message-passing layer (as in PVM).
- The ability to seamlessly use all system resources without modifying the operating system or application code.

15) In this respect, vSMP addresses a need that has long been felt in the field of high-performance computing (HPC): How to achieve the computing performance level of a supercomputer without high-cost dedicated hardware and special-purpose software? vSMP solves this problem by enabling multiple, generic, low-cost computers to be coupled together efficiently to create a single (virtual) high-power machine, in the manner that is described and claimed in the Application. Until very recently, VMWare virtualization was not even used in HPC, and its use in this area is still very limited due to the overhead of the virtualization process. vSMP has overcome these limitations and is rapidly growing in use and recognition.

16) The unique advantages of the vSMP solution have led to its adoption in HPC products offered by a number of industry leaders. For example, Exhibits B and C, attached hereto, contain product literature describing vSMP-based products that are distributed by Hewlett-Packard and Dell. Other major manufacturers offering vSMP-based products include IBM, Sun Microsystems, Cray, and Silicon Graphics. Exhibit D contains a press release in this regard by Cray.

17) The success of vSMP has led to the emergence of at least one imitator: 3Leaf Systems (Santa Clara, California). The 3Leaf product is described in a white paper entitled, "Next Generation Data Center Environment for HPC," which is attached hereto as Exhibit E.

18) Thus, to conclude, vSMP, as described and claimed in the Application, satisfies a long-felt need in the HPC industry and has enjoyed substantial market acceptance as a result. The success of vSMP has begun to attract imitators to the market. On the basis of these objective indicators, it is clear that the invention claimed in the Application is non-obvious. These secondary considerations are in addition to my analysis above, in which I showed that a person having ordinary skill in the art would anyway have been unable to derive the claimed invention from the Bugnion and Nickel references.

19) I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and conjecture are thought to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application of any patent issued thereon.

Dr. Joseph Landman, Citizen of USA
2433 Woodmont East, Canton, MI 48188

Date: 13-October-2009

EXHIBIT
A

Joseph I. Landman,
2433 Woodmont East,
Canton, MI 48188
landman@scalableinformatics.com

EDUCATION:

Wayne State University, Detroit, Michigan
Department of Physics and Astronomy,
Ph.D. in Physics (Computational Theoretical), December 1997
Dissertation: *A Molecular Dynamics Investigation of Low Temperature Grown GaAs*
Advisor: Dr. Caroline Morgan

Michigan State University, East Lansing, Michigan
Department of Physics and Astronomy,
M.S. in Physics, June 1990

SUNY Stony Brook, Stony Brook, New York
Department of Physics,
B.S. in Physics, Dec 1987

HONORS AND AWARDS:

SGI Excellence in Systems Engineering, 1997
New York Regents College Scholarship 1983

RESEARCH AND PROFESSIONAL EXPERIENCE:

Cofounder of Diagnaid Inc Jan 2009-present
Building business model and plans, investor materials. Business and technology planning, marketing planning, sales planning and projection. Pursuit of funding from private capital markets, SBIR. Private capital market pitches and discussions, due diligence processes.

Cofounder of Micass LLC/Navivus Inc Jan 2005-Jan 2009
Building business model and plans, investor materials. Business and technology planning, marketing planning; sales planning and projection. Pursuit of funding from private capital markets, SBIR, MTTC. Prototyping efforts, proofs of concept. Market research on life science computing, medical image processing and related markets. Marketing events, one-on-many presentations and discussions. Private capital market pitches and discussions, due diligence processes.

Research Assistant Professor, Wayne State Jan 2005-June 2007
Teaching high performance computing programming methods to graduate students. Working with a research group on medical image processing tools, and applications around volumetric data navigation. Invention disclosure and due diligence on a novel and intuitive navigation method for clinicians. Began patent application process.

Founder and CEO of Scalable Informatics Inc. Aug 2002-present
Designing and building scalable storage and computing systems. Business development and marketing activities associated with growing a business. Partnership management, technical support, and engineering/research efforts.

Senior Scientist, Bioinformatics, MSC Software Apr 2001-Aug 2002
Architected a distributed parallel computing environment for Bioinformatics applications (MSC.LIFE^{TM}). Built a technical group responsible for the implementation of this product. Educated customers on the product. Performed benchmarking functions to demonstrate scalability. Designed and implemented cluster hardware for internal research and development efforts as well as for customers. Problem resolution efforts, documentation, and analysis. Collaboration with various customers on research projects of mutual interest, including distributed data mining, bioinformatics visualization and information representation.

Systems Engineering Specialist, SGI Apr 1995-Apr 2001
Architected and implemented numerous applications for internal and customer use, including some SGI product (SGI GenomeCluster^{TM}, a scalable bioinformatics computing resource) and pre-cursors (Roboinst derived from my Autoinst). Built collaborations with research universities and individual research groups. Parallelized codes for shared and distributed memory machines. Benchmarked large scale computational systems. Researched performance issues with application scaling for a variety of architectures. Authored papers with collaborators. Collaborated on educational initiatives including Wayne State University's IGERT for Computational Science, and the associated institute ISC. Gave talks at scientific meetings and trade group meetings. Consulted on high performance computing issues for SGI research and educational customer base. Helped customers with algorithm re-design for high performance, re-implementation, and general assistance with coding issues. Performed large scale visualization of proteomics data. Performed proof-of-concept research for customers, including data mining on 10^9 record databases.

Consultant Jan 1992-Apr 1995
IT and infrastructure consultant for a variety of customers.

Graduate Research and Teaching Assistant Sept 1990-Apr 1995
Investigated physical and chemical models using simulation techniques on

supercomputers. Built programs to visualize molecular dynamics trajectories, analyse large runs, recover from problems within the runs. Designed algorithms to compute volumetric charge distribution from modified self-consistent *ab-initio* molecular dynamics code. Parallelized code for SMP SGI machines, vectorized code for Cray machines. Developed new programs and anlyses by encapsulating existing molecular dynamics codes within method/procedure calls to provide a Murnaghan equation of state data fit. Built a code to generate an STM image from a slice through the supercell volume. Managed students, computers, and information access/security. Taught lecture courses, as well as lab and recitation sections of lecture courses.

Graduate Research and Teaching Assistant Sept 1988-Jun 1990
Investigated physical models using simulation techniques on mini-computers. Built programs to visualize physical phenomenon, and measurements. Taught lecture courses, as well as lab and recitation sections of lecture courses.

Associate Engineer, IBM T.J. Watson Research Center Jan 1988-Aug 1988
Performed measurements of onset temperature and critical current density for high Tc superconductors. Built programs to automate measurements using IEEE 488/GPIB bus. Built programs to automate data analysis, reporting, and identify anomolous results for further study. Removed data collection bottleneck resulting in the experiment frequency being dominated by the cryogenic system cooling rates.

Undergraduate Research Assistant Jun 1984-Dec 1987
Worked on developing a computer control for the laser cooling of atomic beams experiment at SUNY and the National Institute of Science and Technology (The PI of this effort at the NIST won the 1997 Nobel prize in Physics for this research, see http://www.nist.gov/public_affairs/releases/n97-26.htm).

TEACHING EXPERIENCE:

Scalable Informatics: sales and marketing training, courses on high peformance computing programming

MSC Software: sales and marketing training. End user training. HPC tutorials.

SGI: sales and marketing training. End user training. HPC tutorials. Data Mining for bioinformatics research.

Wayne State: Introductory (calculus based) Physics lecture, recitations, and laboratories. Advanced laboratory sections in modern physics experiments.

Recitation sections for non-calculus based physics.

Michigan State University: Advanced laboratory section for modern physics experimental course.

PATENTS:

- USPTO #7,249,357 "Transparent distribution and execution of data in a multiprocessor environment "

PUBLICATIONS:

- **"Accelerating HMMer searches on Opteron processors with minimally invasive recoding",** Joseph I. Landman, Joydeep Ray, J. P. Walters: AINA (2) 2006: 628-636

- **"Parallelization of a Legacy Program using OpenMP",** Landman J. and Piecuch P., ACM FORTRAN Forum, 19, 2, August 2000, 16-23.

- **"Parallelization of a multi-reference coupled-cluster method"** Piotr Piecuch and Joseph I. Landman, *Parallel Computing*, Vol 26, 7-8, July 2000

- **"Arsenic Interstial Pairs in GaAs"** P. Papoulis, C.G. Morgan, J.T. Schick, J.I. Landman, and N. Rahhal-Orabi, Materials Science Forum **258-263**, 923 (1997).

- **"Arsenic Interstials and Interstitial Complexes in Low-Temperature-Grown GaAs"** J.I. Landman, C.G. Morgan, J.T. Schick, P. Papoulis, and A. Kumar, Phys. Rev. B **55**, 15581 (1997).

- **"Arsenic-Antisite-Related Defects in GaAs Grown at Low Temperatures: Characterization of Localized States"** J.I. Landman, C.G. Morgan, J.T. Schick, A. Kumar, P. Papoulis, and M.F. Kramer, Materials Science Forum **196-201**, 249 (1995).

- **"Antisite-Related Defects in GaAs Grown at Low Temperatures"** J.I. Landman, C.G. Morgan, and J.T. Schick, Phys. Rev. Lett. **74**, 4007 (1995).

- **"Interstitial Defects in II-VI Semiconductors: Role of the Cation d States"** J.T. Schick, C.G. Morgan, and J.I. Landman, Materials Science Forum **83-87**, 1253 (1992)

PRESENTATIONS:

- "**The unreasonable effectiveness of clusters for life science and informatics computing**", ClusterWorld 2004 invited talk

- "**Building Software for High Performance Informatics and Chemistry**", ClusterWorld 2003 invited talk

- **“Bio-IT: the nuts and bolts”,** Eastern Michigan University Center for Entrepreneurship and Michigan Center for Biological Information, Jan. 2003

- **“Scalability Matters - Why We Need to Make Bioinformatics Programs Scalable, and Results from Work on Various Programs”,** Michigan State University, Center for Biological Modeling, Oct. 2001

- **“The DNA of Computing”** Intel invited talk to LinuxWorld August 2001, on High Performance Informatics,

- **Building Optimal Computational Environments for Bioinformatics and Computational Chemistry”,** SGI Bioinformatics Seminar, ISMB 2000, August 2000

- **“Linux for Software Development**”, SGI Linux University, March – July 2000

- **“Linux Clusters for Computational Science**”, Semana de Supercómputo, UNAM Departamento de Supercómputo , May 2000

BOARD MEMBERSHIP:

Honorary (Emeritus) Director: Bioinformatics.org